Claims

[c1] 1.A method for testing an integrated circuit, the method comprising:
grouping a circuit logic into one or more logic areas;
associating each one of the logic areas with one or more first testcases;
altering a circuit in one of the logic areas to create an altered logic area from an unaltered said logic area;
retesting the altered logic area using only with the one or more first testcases associated with the unaltered logic area;
identifying failed first testcases; and
counting the number of logic areas in the failed first testcases to predict which logic areas are adversely affected by the altering of the circuit.

[c2] 2.The method of claim 1, further comprising:
identifying other logic area utilized by the filed first testcases, the other logic area being different from the altered logic area;
re-running at least one second testcase that tests the other logic areas;
identifying the second testcases that fail; and

identifying common logic areas affected by the failed second testcases and the failed first testcases to predict which logic areas are adversely affected by the altering of the circuit.

[c3] 3.The method of claim 2, wherein the failed first and second testcases are mapped on a scoreboard that affords a visual view of a relationship among affected logic areas to predict which logic areas are defective.

[c4] 4.The method of claim 1, wherein the predicting of which logic areas are adversely affected is achieved by identifying a common logic area in both the first and second failed testcases.

[c5] 5.The method of claim 1, wherein the integrated circuit is a virtual circuit created by a hardware descriptor language.

[c6] 6.The method of claim 1, further comprising:
running a new testcase on the circuit;
if the new testcase fails, examining other failed first testcases; and
identifying a common logic area in the failed new testcase and the failed first testcases to predict a defect in the common logic area.

[c7] 7.A system for testing an integrated circuit, the system

comprising:
means for grouping a circuit logic into one or more logic areas;
means for associating each one of the logic areas with one or more first testcases;
means for altering a circuit in one of the logic areas to create an altered logic area from an unaltered said logic area;
means for retesting the altered logic area using only with the one or more first testcases associated with the unaltered logic area;
means for identifying failed first testcases; and
means for counting the number of logic areas in the failed first testcases to predict which logic areas are adversely affected by the altering of the circuit.

[c8] 8.The system of claim 7, further comprising:
means for identifying other logic area utilized by the filed first testcases, the other logic area being different from the altered logic area;
means for re-running at least one second testcase that tests the other logic areas;
means for identifying the second testcases that fail; and
means for identifying common logic areas affected by the failed second testcases and the failed first testcases to predict which logic areas are adversely affected by the

altering of the circuit.

[c9] 9.The system of claim 8, wherein the failed first and second testcases are mapped on a scoreboard that affords a visual view of a relationship among affected logic areas to predict which logic areas are defective.

[c10] 10.The system of claim 7, wherein the predicting of which logic areas are adversely affected is achieved by identifying a common logic area in both the first and second failed testcases.

[c11] 11.The system of claim 7, wherein the integrated circuit is a virtual circuit created by a hardware descriptor language.

[c12] 12.The system of claim 7, further comprising:
means for running a new testcase on the circuit;
means for, if the new testcase fails, examining other failed first testcases; and
means for identifying a common logic area in the failed new testcase and the failed first testcases to predict a defect in the common logic area.

[c13] 13.A computer program product, residing on a computer usable medium, for testing an integrated circuit, the computer program product comprising:
program code for grouping a circuit logic into one or

more logic areas;
program code for associating each one of the logic areas with one or more first testcases;
program code for altering a circuit in one of the logic areas to create an altered logic area from an unaltered said logic area;
program code for retesting the altered logic area using only with the one or more first testcases associated with the unaltered logic area;
program code for identifying failed first testcases; and
program code for counting the number of logic areas in the failed first testcases to predict which logic areas are adversely affected by the altering of the circuit.

[c14] 14.The computer program product of claim 13, further comprising:
program code for identifying other logic area utilized by the filed first testcases, the other logic area being different from the altered logic area;
program code for re-running at least one second testcase that tests the other logic areas;
program code for identifying the second testcases that fail; and
program code for identifying common logic areas affected by the failed second testcases and the failed first testcases to predict which logic areas are adversely af-

fected by the altering of the circuit.

[c15] 15.The computer program product of claim 14, wherein the failed first and second testcases are mapped on a scoreboard that affords a visual view of a relationship among affected logic areas to predict which logic areas are defective.

[c16] 16.The computer program product of claim 13, wherein the predicting of which logic areas are adversely affected is achieved by identifying a common logic area in both the first and second failed testcases.

[c17] 17.The computer program product of claim 13, wherein the integrated circuit is a virtual circuit created by a hardware descriptor language.

[c18] 18.The computer program product of claim 13, further comprising:
program code for running a new testcase on the circuit;
program code for, if the new testcase fails, examining other failed first testcases; and
program code for identifying a common logic area in the failed new testcase and the failed first testcases to predict a defect in the common logic area.